10

(UDF) 2.50. A document entitled "White paper Blu-ray Disc™ Read-Only Format—2.B Audio Visual Application Format Specifications for BD-ROM Version 2.4" available from http://www.blu-raydisc.com/Assets/Downloadablefile/ BD-ROM_Audio_Visual_Application_Format_Specifica-tions-18780.prf (the disclosure of which is incorporated by reference herein in its entirety) explains that a BD-ROM can be authored in High Definition Movie (HDMV) mode or BD-J mode. The HDMV mode is intended as an easy-to-author framework supporting all of the features supported by the DVD-Video specification. Indeed, HDMV mode is intended for use during the content production and material preparation process of authoring DVD-Video interactive multimedia content to enable simultaneous production of DVD-Video and BD-ROM content. The BD-J mode is based on the Java™ platform and provides a fully programmable application environment with network connectivity. Java is a platform independent programming environment that is deployed in a wide variety of environments.

[0108] The whitepaper referenced above provides details concerning the BD-ROM file format including an illustration of the BD-ROM file format reproduced as FIG. 9. The file system 160 includes a BDMV directory 162 that contains the PLAYLIST 164, CLIPINF 166, STREAM 168, AUXDATA 170, and BACKUP 172 directories. The PLAYLIST directory 164 contains the database files 174 for movie playlists. Each playlist has its own xxxxx.mpls file. The CLIPINF directory 166 contains the database files 176 for clips. The STREAM directory 168 contains audio and video stream files 178 (e.g. BDAV MPEG-2 transport streams) and the SSIF directory 180, which contains stereoscopic interleaved files 182. The AUXDATA directory 170 contains sound data files 184 and font files 186. The BACKUP directory 172 contains copies of the "index.bdmv file", the "MovieObject.bdmv" file, all the files in the PLAYLIST directory 164 and all files in the CLIP-INF directory 166. The BDMV directory 162 also contains the index.bdmv file 188 and the MovideObject.bdmv file 190. The index.bdmv stores information describing the contents of the BDMV directory. The MovieObject.bdmv file stores information for one or more Movie Objects.

[0109] The white paper also provides a simplified overview of the BD-ROM data structure, which is reproduced as FIG. 10. The BD-ROM data structure 200 includes an Index Table 202 at the top-level of the data structure, which defines the Titles and the Top Menu of a BD-ROM disc. An Index Table is conceptually illustrated in FIG. 11. The Index Table 202 contains entry points for all of the Titles and the Top Menu and has an entry to a Movie Object 222 and/or a BD-J Object 224 designated for "First Playback" 226. When the disc is loaded, the player refers to the "First Playback" entry to determine the corresponding Movie Object or BD-J Object that shall be executed. The Index Table 202 also contains an entry 228 for "Top Menu" and an entry 230 for each title.

[0110] Referring back to FIG. 10, the layer below the Index Table includes the Movie Objects 204 and the BD-J objects 206. Each Movie Object is an executable navigation command program. The navigation commands can launch PlayList playback or another Movie Object enabling the authoring of a set of Movie Objects for managing playback of Playlists in accordance with a user's interaction and preferences. BD-J objects 206 perform similar functions, but are implemented as Java Xlet.

[0111] The Movie Playlist layer 208 includes a plurality of Movie Playlists. A Movie Playlist is a collection of playing intervals in the Clips. One such interval is referred to as a PlayItem and includes an IN-point and an OUT-point, each of which refers to positions on a time axis of the Clip. The manner in which MoviePlaylists index into the clips is illus-trated in FIG. 12. Each Movie Playlist 230 includes IN-points 232 and OUT-points 234 that index to positions in a Clip 236.

[0112] Referring back to FIG. 10, the bottom layer of the BD-ROM data structure is the Clip layer 210. An AV stream file together with its associated database attributes is consid-ered to be one object. The AV stream file is called a Clip AV stream file 212, and the associated database attribute file is called a Clip Information File 214. The Clip AV stream file 212 stores an MPEG-2 Transport Stream in accordance with ISO/IEC 13818-1 in a structure compliant with the BD-ROM AV specification. The Clip Information file stores the time stamps of the access point into the corresponding AV stream file. A player can read the Clip Information file to find out the position where it should begin to read the data from the AV stream file.

[0113] The Clip AV stream file can include an Interactive Graphics stream that is utilized in HDMV to generate a "Pop-Up" Menu Interface. In this case, video playback can con-tinue while the HDMV Interactive Graphics are on the screen or video playback may be paused. Using the HDMV Interac-tive Graphics framework, multi-page menus can be defined with special commands available for inter-page navigation. As part of the framework, Button Objects 216 can be defined. When a Button is activated, a corresponding navigation com-mand is executed which causes the display to change to a specified page. The HDMV Interactive Graphics framework also provides a scheme for dynamic graphics display. On a single page, this enables the Content Provider to determine dynamically which Buttons are visible and invisible at any point in time.

[0114] For HDMV BD-ROM content, Movie Objects and the Button Objects contain navigation commands. A Movie Object is executed when the Title associated with the Movie Object begins playback. Movie Object navigation commands are used to manage PlayList playback. While a PlayList is under playback, the state of a Movie Object is maintained. A Button Object is an alternative programming method that is available while the PlayList is under playback and a Button Object is executed by user activation or system timer.

Building an Object Model of a HDMV BD-ROM

[0115] Using processes similar to those described above with respect to interactive multimedia content authored in accordance with the DVD-Video specification, content authored in accordance with the BD-ROM specification uti-lizing the HDMV framework can be parsed to create an object model. The Index Table can be parsed to identify each of the Movie Objects. The Movie Objects can then be used to insert objects corresponding to the Movie Playlists, Clips, and But-ton Objects into the object model utilizing the navigation information associated with the Movie Objects and the But-ton Objects. Once the object model has been constructed, the object model can be utilized to generate an HTML5 page for each Movie Object and an associated JavaScript file that captures the navigation information contained within Movie Objects and the Button Objects. In addition, audio/video/ subtitle information in the clips associated with each Movie Object can be extracted, transcoded (optional) and inserted into one or more container files in a manner similar to that